REMARKS

This is in response to the Office Action of May 13, 2008, in which all of pending claims 1-21 were rejected under 35 U.S.C. § 103(a) as unpatentable over U.S. Patent No. 7,085,805 to Ruberg et al. in view of U.S. Patent No. 7,000,037 to Rabinovitz et al. and U.S. Patent No. 7,190,477 to Ferlitsch.

Embodiments of the invention claimed in the present application, generally speaking, provide systems and methods for deterministic communications of serial data over computer networks by reducing transmission latencies between a network connected computer and a serial device server to a level that approximates the latency of a directly connected serial device. Such near real-time transmission requirements are particularly relevant for time-sensitive applications including, for example, material handling, satellite communications, and real-time device monitoring. Embodiments according to the present invention employ several techniques for reducing latency delays. Semi-blocking read functions for getting data as soon as it is available, push to FIFO functions for writing data to the FIFO before writing the data to the queue for each serial device, and intercharacter timeout functions each serve to reduce latency regardless of the transport protocol. Additional efficiencies may be achieved by eliminating redundant error checking, such as through custom transport protocols. For dedicated LAN environments with extreme sensitivity to latency delays, the custom transport protocol offers the best performance; however, for most applications, TCP is adequate.

Independent claim 1 recites a device server system including serial devices and a device server. The device server connects the serial devices to a local area network. Driver software is installed on a host computer with an operating system that is connected to a local area network. The driver software directs serial port data onto the local area network. Firmware is installed on the device server for facilitating communication between the host computer and a selected one of the serial devices. **The firmware directs the serial port data received from the remote computer to the selected one of the serial devices, and writes portions of the received serial port data to FIFO registers of the serial device, if a queue is empty, before writing**

remaining data of the received serial port data to the queue of the selected one of the serial devices.

Independent claim 8 recites a system for connecting serial devices to a local area network. A device server is in communication with the local area network. The device server includes serial connectors for linking one or more serial devices to the device server, an operating system, and applications running on the operating system for coordinating communications between the operating system, the local area network, and the serial device. A driver is installed on a host computer for mediating between host application software and host operating system software installed on the host computer. The host computer is in communication with the local area network. The server driver transmits serial data from the host application software over the local area network to a selected serial device linked via one of the serial connectors to the device server. **The applications have a semi-blocking read function for transferring the serial data from a memory queue associated with the selected serial device to a buffer of the operating system.**

Independent claim 13 recites a system for connecting serial devices to a network. Host driver software is installed on a host computer connected to the network. The host driver software interfaces with host application software and host operating system software. The host operating system software has a serial port API with an intercharacter interval timer setting. The host driver software receives serial port data over the network using the serial port API. A device server is connected to the network and in electrical communication with one or more serial devices. **The device server sends the serial port data over the network and measures an intercharacter interval corresponding to the intercharacter interval timer setting of the serial port API, and returns a flag to the host computer when the intercharacter interval exceeds a preset value.**

Independent claim 14 recites a method for facilitating communication between a host computer and a serial device over a network. Data is transmitted in a standard serial port format over a local area network to a selected serial device having a FIFO register and a queue.

The data is received at a device server connected to the network. The device server has one or more serial ports connected to one or more serial devices. One of the one or more serial devices is the selected serial device. The received data is stored temporarily in a buffer. The received data is read from the buffer. **The read data is written to the selected serial device, such that if the FIFO register is empty, the read data is written to the FIFO register until the FIFO register is full and the remaining data is written to the queue.**

Independent claim 19 recites a method for eliminating synchronization latencies in a device server for facilitating communication between a host computer and a selected serial device. **Data and a parameter for a desired intercharacter timer interval are transmitted in a standard serial port format over a network link to a remote device server.** The remote device server receives the data and the parameter. The data is written to the selected serial device. **A time interval of the remote device server is measured. A flag is returned to the host computer if the time interval exceeds the parameter.**

Independent claim 20 recites a method for facilitating communication between a host computer and a remote serial device. Serial data from an application on the host computer is transmitted over a network link. A device server receives the serial data from the network link. The serial data is stored temporarily in a buffer. **The serial data is transferred from the buffer to the remote serial device using a semi-blocking read mode function.**

Ruberg et al. is directed to distributed computer networks in which server computers are responsible for managing communications to and from peripheral devices (e.g. keyboards, mice, monitors) connected to so-called "thin clients" that are so "thin" the devices do not have enough space to store and process serial device drivers. *See generally* Ruberg et al. col. 2, lines 18-22 ("The invention relates to methods and apparatus for data distribution among servers in a grouped server system where device operations remain uninterrupted when a server fails. Servers in the grouped server system do not pose a single point of failure and implementation is relatively simple."). In particular, Ruberg et al. is directed to systems that distribute the management of the serial devices (peripheral devices connected to thin clients) among more than

one server computer. *Id.* at col. 2, lines 34 ("In accordance with the invention, a plurality of servers operate as a group. Each server in the group includes a device manager which is responsible for brokering devices to services. A service communicates to a device manager on the same server through an inter-process communication mechanism. A desktop unit coupled to a server maintains persistent connections with a single device manager. When the desktop unit is reset or when a device manager the desktop unit is connected to fails, the desktop unit switches to a peer (other device managers in the group) or reconnects with the same device manager if the device manager recovers prior to establishing a new connection with a peer.").

Rabinovitz et al., as the title "LARGE ARRAY OF MASS DATA STORAGE DEVICES CONNECTED TO A COMPUTER BY A **SERIAL** LINK" suggests, is directed to "an arrangement for using **a serial-PCI connection between computing devices and remote mass data storage-devices** in order to create highly accessible, flexible, high-performance, low-cost storage systems." Rabinovitz et al. col. 1, lines 10-14 (emphasis added).

Ferlitsch is directed to a networked printing system in which print data is despooled directly from printing sources (i.e. client computers on the network) to the printing device based on instructions contained in a print job ticket. In network printing environments, it is common to despool print data from a client computer to a print server computer, and then despool the data from the print server to the printer. The invention disclosed in Ferlitsch reduces network traffic by removing one of the despooling operations, i.e. despooling the data from the print server to the printer, from the process.

None of claims 1-21 are rendered obvious by Ruberg et al. in view Rabinovitz et al. and Ferlitsch, because each claim requires at least one element that is not taught or suggested by the references, alone or in combination, and that would not have been obvious to an ordinarily skilled artisan with knowledge of the three references. Independent claim 1 requires a device server with firmware that directs serial port data received from a remote computer to a selected one of a number of serial devices, and **writes portions of the received serial port data to FIFO registers** of the selected serial device, **if a queue is empty, before writing remaining data of the**

received serial port data to the queue of the selected one of the serial devices. Ruberg et al. does not teach any particular method or disclose any system for selectively writing data to the FIFO registry and queue of a serial device. In fact, Ruberg et al. never mentions FIFO registries (or any equivalent structure or function) on serial devices, let alone discloses or teaches the specific limitation required by claim 1. Similar to Ruberg et al., Rabinovitz et al. never once mentions FIFO registries on serial devices, or any equivalent structure or function. Ferlitsch discusses FIFO registries but only in the context of one of several standard methods of "despooling" print data from a job ticket queue. Ferlitsch therefore fails to teach or suggest, or somehow render obvious the specific elements of claim 1 that are missing from Ruberg et al. and Rabinovitz et al. Therefore, the references relied on in the Office Action do not, alone or in combination, disclose, teach, or suggest a device server with firmware that writes portions of the received serial port data to FIFO registers of the selected serial device, if a queue is empty, before writing remaining data of the received serial port data to the queue of the selected one of the serial devices. Additionally, the references, in any combination, do not somehow inherently teach or otherwise render obvious without explicitly or implicitly disclosing the limitations of claim 1 that are emphasized in bold print above. Claims 2-7 depend from claim 1 and are allowable therewith.

Independent claim 8 requires a device server with **applications that have a semi-blocking read function for transferring serial data from a memory queue associated with the selected serial device to a buffer of the operating system**. The "semi-blocking" read function reduces latency because the call returns data immediately when it becomes available without waiting for a specified byte count or a polling period, such as is the case in blocking and non-blocking read methods respectively. *See* U.S. Patent App. Serial No. 10/788,959 p. 10, line 22 – p. 11, line 6. Ruberg et al., Rabinovitz et al., and Ferlitsch, alone or in combination, do not teach any particular method or disclose any system that uses a semi-blocking read function for transferring serial device data. In fact, none of the references even mention the concept of blocking or non-blocking read functions, let alone disclose or teach the specific semi-blocking read limitation required by claim 8. Therefore, the references relied on in the Office Action do not,

alone or in combination, disclose, teach, or suggest a device server with applications that have a semi-blocking read function for transferring serial data from a memory queue associated with the selected serial device to a buffer of the operating system. Additionally, the references, in any combination, do not inherently teach or otherwise render obvious (without explicitly or implicitly disclosing) the limitations of claim 8 that are emphasized in bold print above. Claims 9-12 depend from claim 8 and are allowable therewith.

Independent claim 13 requires a device server that **sends serial port data over a network and measures an intercharacter interval corresponding to an intercharacter interval timer setting of a serial port API on a host operating system, and returns a flag to the host computer when the intercharacter interval exceeds a preset value**. In embodiments of the present invention, the device server times the data received from the serial device. When an inter-character interval is detected that exceeds the inter-character timeout setting, a timeout event is sent to the server driver (on the host computer) along with the serial data. The serial data and timeout events are sent in a sequence such that the server driver is able to terminate application requested read operations at the correct points in the data stream. This eliminates the errors in inter-character timeout detection that had been introduced by network latency in configurations where the host computer not the device server performed the intercharacter interval measurements. *See* U.S. Patent App. Serial No. 10/788,959 p. 11, line 28 – p. 12, line 16. Ruberg et al., Rabinovitz et al., and Ferlitsch, alone or in combination, do not teach any particular method or disclose any system for handling intercharacter interval measurements or timeouts. In fact, none of the references even mention the concept of intercharacter intervals, let alone disclose or teach the specific limitation required by claim 13. Therefore, the references relied on in the Office Action do not, alone or in combination, disclose, teach, or suggest a device server that sends serial port data over a network and measures an intercharacter interval corresponding to an intercharacter interval timer setting of a serial port API on a host operating system, and returns a flag to the host computer when the intercharacter interval exceeds a preset value. Additionally, the references, in

any combination, do not inherently teach or otherwise render obvious (without explicitly or implicitly disclosing) the limitations of claim 13 that are emphasized in bold print above.

Independent claim 14 recites a method including the step of writing data read from a buffer on the device server to a selected serial device, **such that if the FIFO register is empty, the read data is written to the FIFO register until the FIFO register is full and the remaining data is written to the queue**. Ruberg et al. does not teach any particular method or disclose any system for selectively writing data to the FIFO registry and queue of a serial device. In fact, Ruberg et al. never mentions FIFO registries (or any equivalent structure or function) on serial devices, let alone discloses or teaches the specific limitation required by claim 14. Similar to Ruberg et al., Rabinovitz et al. never once mentions FIFO registries on serial devices, or any equivalent structure or function. Ferlitsch discusses FIFO registries but only in the context of one of several standard methods of "despooling" print data from a job ticket queue. Ferlitsch therefore fails to teach or suggest, or somehow render obvious the specific elements of claim 14 that are missing from Ruberg et al. and Rabinovitz et al. Therefore, the references relied on in the Office Action do not, alone or in combination, disclose, teach, or suggest a method (or system for carrying out a method) including the step of writing data read from a buffer on the device server to a selected serial device, such that if the FIFO register is empty, the read data is written to the FIFO register until the FIFO register is full and the remaining data is written to the queue. Additionally, the references, in any combination, do not somehow inherently teach or otherwise render obvious without explicitly or implicitly disclosing the limitations of claim 1 that are emphasized in bold print above. Claims 15-18 depend from claim 14 and are allowable therewith.

Independent claim 19 recites a method including the steps of **transmitting** data in a standard serial port format and **a parameter for a desired intercharacter timer interval over a network link to a remote device server, measuring a time interval of the remote device server**, and **returning a flag to the host computer if the time interval exceeds the parameter**. Ruberg et al., Rabinovitz et al., and Ferlitsch, alone or in combination, do not teach any particular method or disclose any system for handling intercharacter interval measurements or timeouts. In

fact, none of the references even mention the concept of intercharacter intervals, let alone disclose or teach the specific limitation required by claim 13. Therefore, the references relied on in the Office Action do not, alone or in combination, disclose, teach, or suggest a method (or system for carrying out a method) including the steps of transmitting data in a standard serial port format and a parameter for a desired intercharacter timer interval over a network link to a remote device server, measuring a time interval of the remote device server, and returning a flag to the host computer if the time interval exceeds the parameter. Additionally, the references, in any combination, do not inherently teach or otherwise render obvious (without explicitly or implicitly disclosing) the limitations of claim 19 that are emphasized in bold print above.

Independent claim 20 recites a method including the step of **transferring serial data from a buffer on a device server to a remote serial device using a semi-blocking read mode function**. Ruberg et al., Rabinovitz et al., and Ferlitsch, alone or in combination, do not teach any particular method or disclose any system that uses a semi-blocking read function for transferring serial device data. In fact, none of the references even mention the concept of blocking or non-blocking read functions, let alone disclose or teach the specific semi-blocking read limitation required by claim 20. Therefore, the references relied on in the Office Action do not, alone or in combination, disclose, teach, or suggest a method (or system for carrying out a method) including the step of transferring serial data from a buffer on a device server to a remote serial device using a semi-blocking read mode function. Additionally, the references, in any combination, do not inherently teach or otherwise render obvious (without explicitly or implicitly disclosing) the limitations of claim 20 that are emphasized in bold print above. Claim 21 depends from claim 20 and is allowable therewith.

Prima facie obviousness is a procedural tool of examination which allocates who has the burden of producing evidence in each step of the examination process. Manual of Patent Examining Procedure § 2142. The **examiner bears the initial burden of factually supporting** any *prima facie* conclusion of obviousness. *Id.* The applicant is under no obligation to submit evidence of nonobviousness if the examiner does not establish *prima facie* obviousness. *Id.* To

support a rejection under 35 U.S.C. § 103, the **reason(s)** why the claimed invention would have been obvious **must be clearly articulated**. *Id.* The Supreme Court has noted that the analysis supporting an obviousness rejection should be made explicit. *KSR International Co. v. Teleflex Inc.*, 550 U.S. ___, ___, 82 USPQ2d 1385, 1396 (2007). The Federal Circuit has stated that "rejections on obviousness **cannot be sustained with mere conclusory statements**; instead, **there must be some articulated reasoning with some rational underpinning** to support the legal conclusion of obviousness." *In re Kahn*, 441 F.3d 977, 988, 78 USPQ2d 1329, 1336 (Fed. Cir. 2006) (emphasis added). "**All words in a claim must be considered** in judging the patentability of that claim against the prior art." M.P.E.P. § 2143.03 (quoting *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970)).

The Office Action fails to establish *prima facie* obviousness, because it does not articulate any rational reason(s) why, having considered all the claimed limitations, the references relied on render any of independent claims 1, 8, 13, 14, 19 and 20, let alone all pending claims 1-21 obvious. The Office Action relies on Ruberg et al. as the primary reference, but in doing so refers to what appears to be a consolidated characterization of the invention embodied in the different claims pending in the present application. Unfortunately, the **actual claim language** is not considered with respect to any of the references nor does the Office Action explain where or how they teach or suggest the invention embodied in the claim language of any of the claims. Furthermore, the cited references, alone or in the manner combined in the Office Action, do not teach or suggest at least one element of each of independent claims 1, 8, 13, 14, 19 and 20 and therefore cannot render any of the claims obvious under 35 U.S.C. § 103(a).

The Office Action states that Ruberg et al. teaches a device with firmware that translates "between Internet network messages and serial (USB) messages." Office Action of May 13, 2008, p. 2. However, this language is not found in any of the claims and it is not even clear from the Office Action which particular claims it is meant to relate to for the purposes of the rejection. While it is true that the claims of the present application relate generally to subject matter that includes network communications and serial data communications, the quoted

language from the Office Action bears no direct relevance to the claims and does not form a proper basis for finding that any limitation is taught by Ruberg et al. The Office Action also states that the system in Ruberg et al. "may handle timeouts and intercharacter interval flags." *Id.* While both timeouts and intercharacter interval flags are terms used in the claims, the Office Action provides no reason why Ruberg et al. teaches the actual claim limitations. Rather, the Office Action appears to simply state that Ruberg et al. refers to these concepts in general, which, of course, is not a proper basis for rejecting any claim. Furthermore, the excerpt cited in the Office Action as supporting the assertion that Ruber et al. teaches these concepts refers to heartbeat messages, not intercharacter interval timouts or flags. In fact, Ruberg et al. never once mentions intercharacter intervals.

The Office Action goes on to combine Ruberg et al. with Rabinovitz et al. by stating that Ruberg et al. "does not expressly state the particular method of Ethernet-serial conversion, including usage of cables and ports", but that Rabinovitz et al. teaches a system where the "serialization and placement of information is more defined." *Id.* It is not clear to Attorneys for Applicant what, if any, relevance these statements bear on the claims of the present application. "Ethernet-serial conversion," let alone such conversions using "cables and ports" is not part of any limitation of any claim in the present application. Thus, the reason for combining Ruberg et al. and Rabinovitz et al. asserted, i.e. what is missing from Ruberg et al. that is taught by Rabinovitz et al. is not relevant to any of the claims and therefore the combination does not form a proper basis for rejecting any of the claims. Additionally, and leaving aside whether Rabinovitz et al. actually teaches the limitation, a system where "serialization and placement of information is more defined" is also not part of any claim of the present application. Thus, even if the combination of Ruberg et al. and Rabinovitz et al. formed a proper basis for rejection, the alleged teaching of Rabinovitz et al., i.e. a system where "serialization and placement of information is more defined" does not relate in any way to any claim in the present application.

The combination of Ruberg et al. and Ferlitsch in the Office Action is similarly flawed as Ruberg et al. with Rabinovitz et al. *Id.* at p. 3. The Office Action states that while

Ruberg et al. does not disclose the particulars of queuing, using FIFO registers and serial chips, Ferlitsch teaches a "system (abstract) of providing access to shared serial devices over a network ... that teaches these limitations." *Id.* The question is, what limitations does Ferlitsch teach? Several claims use one of the term queue and the term FIFO register, but it is not clear which if any specific claim or claim limitation(s) the Office Action purports to apply Ferlitsch. Furthermore, Ferlitsch does not disclose anything with regard to, as an example, FIFO registers that could be construed as teaching any limitation of any claim. In fact, the only mention of FIFO is by way of explaining one algorithmic method for despooling print data to printers from a print job ticket queue that in no teaches or suggests any claim limitation in the present application. *See, e.g.*, Ferlitsch, col. 12, lines 40-45 ("Print server spooler 164 authorizes the despooling of print data to printing device(s) 108 **by removing the associated print job ticket from the job ticket print queue 168 by some algorithm, such as FIFO**, or by size/aging, and may consider priority considerations from the printing source 80") (emphasis added).

Therefore, the Office Action fails to make a *prima facie* case of obviousness because it fails to relate the references cited as prior art to any of the actual language of the claims of the present application. Furthermore, even assuming, without admitting, that a *prima facie* case of obviousness has been established, none of pending claims 1-21 are rendered obvious under 35 U.S.C. § 103(a) because each claim requires at least one element that is not taught or suggested by the cited references, alone or in combination, and that would not have been obvious to an ordinarily skilled artisan with knowledge of the three references. As each of pending claims 1-21 requires at least one element not found in the prior art and not within the knowledge of an ordinarily skilled artisan, the combination recited in each claim is also not obvious under 35 U.S.C. § 103(a) over Ruberg et al. in view Rabinovitz et al. and Ferlitsch.

It should be noted that while each of the references relied on in the Office Action have been discussed separately in this Response, that does not mean that they have been analyzed individually for purposes of the patentability of the claims under 35 U.S.C. § 103(a). The distinctions drawn between a particular reference and the claims apply to that reference and the

reference combined as suggested in the Office Action. In other words, it is not accurate to state that any of the above arguments are incorrect because, for example, Ruberg et al. has been considered individually instead of in combination with either Rabinovitz et al. or Ferlitsch. The references have been considered individually **and in combination** with the other references relied on in the Office Action and still fail to teach or suggest the subject matter recited in and thereby fail to render obvious any of pending claims 1-21.

Based on the foregoing remarks, none of claims 1-21 are rendered obvious under 35 U.S.C. § 103(a) by Ruberg et al. in view of Rabinovitz et al. and Ferlitsch and the present application is therefore in condition for allowance and notice to that effect is respectfully requested.

Respectfully submitted,

KINNEY & LANGE, P.A.

Date: November 13, 2008

By: /Alan M. Koenck/

Alan M. Koenck Reg. No., 43,724
THE KINNEY & LANGE BUILDING
312 South Third Street
Minneapolis, MN 55415-1002
Telephone: (612) 339-1863
Fax: (612) 339-6580